

71. (Amended) A data processing system [having] for
analyzing relations between entities of different types, the
system comprising:

data storing means;

recording means for recording in the data storing
means, entity-type defining data identifying each
entity type and entity-instance data identifying
instances of each type of entity [relationships between
objects] , and

entity instance/type associating means for
associating each recorded entity-instance datum with an
entity-type defining datum

where recording is performed such that all entity-
instance data associated with a singular entity-type
defining datum [done using object identifiers which] are
stored in an instance-storing table of uniform [in]
layout

[across all object types and furthermore

where identifiers include the object type
descriptor and an object number].

2. (Amended) The data processing system of Claim 1,
wherein:

the recording means includes relative table means
for recording and retrieving data according to a
relative table organization, and

each entity-type defining datum is stored in a
unique row of an entity-definition table having a
relative table organization so that each entity-type
defining datum is addressable by its relative row
number

[the object number of the object identifier is a
system-key (SYSKEY), relative row number or other such
row address which allows for direct access to the
object data].

3. (Amended) The data processing system of Claim 1, wherein the entity instance/type associating means includes association enforcing means for preventing the recording of a new entity-instance datum in the instance-storing table if the new entity-instance datum is not associated with a pre-existing entity-type defining datum

[objects can be recorded only after a definition has been entered allowing for the object type to exist].

4. (Amended) The data processing system of Claim 2 [3], wherein said relative table means includes table expansion means for adding new entity-type defining data to the entity-definition table without system recompilation so that new [object] entity types can be [entered] defined by a user [based on supplied object classes] at any time during the life of the system [and does not require programmatic changes or] without requiring recompilation of the system.

5. (Amended) The data processing system of Claim 4, where the entity-definition table includes entity-defining data which defines one or more [Object] Classes provided are based on] common entity-types selected from the group consisting of "Address", "Account", "Branch", "Customer" and "Phone No." [entities used in commercial application].

6. (Amended) The data processing system of Claim 10 [1], further comprising mandatory relationship enforcing means for preventing the recording of a relation-instance datum if the relation-instance datum is not associated with a prespecified, relation-type defining datum

[wherein relationship between objects can be recorded only after a definition has been entered allowing for the relationship type to exist between the two or more object types].

7. (Amended) The data processing system of Claim 6, wherein said relative table means includes table expansion means for adding new relation-type defining data to the relation-definition table without system recompilation so that new allowed relationship definitions can be entered by a user at any time during the life of the system [and does not require programmatic changes or] without requiring recompilation of the system.

8. (Amended) The data processing system of Claim 1, further comprising inquiry-path defining means for generating linkages between multiple pieces of entity-instance data, said linkages matching prespecified ones of the relationship-instance data that define relationships between said multiple pieces of entity-instances data thereby allowing [which allows] for inquiries to be formulated to follow existing relationships from one entity type to another within the set of defined allowed relationships[, and for execution of such inquiries].

9. (Amended) The data processing system of Claim 8,
further comprising access-control means for attaching
individual access-control parameters to individual inquiry
paths to thereby limit said inquiry paths accessing data
from a predetermined subset of the entity-instance storing
and relation-instance storing tables so that [which allows]
individual inquiries [to] can have their own access security
independent of other inquiries [and be attached to separate
menus].

10. (Amended) A computer database system comprising:
[in which entities are stored in]
 a memory unit for storing tables of data;
 a finite set of [relative] entity-instances
 storing tables defined in said memory unit [(one for
 each object class)], where each entity-instances table
 is associated with a unique one or more entity type in
 a predefined set of entity types;
 [relationships are stored in]
 a finite set of relation-instances data
 storing tables defined in said memory unit, where
 each relation-instances table is associated with a
 unique one or more relation type in a predefined
 set of relation types [and]
 an entity-type defining table (EntDef)
 defined in said memory unit for storing data
 representing said predefined set of entity types;
 a relation-type defining table (RelDef) defined
 in said memory unit for storing data representing said
 predefined set of relation types; and
 schema defining means for defining a relational
 schema wherein a first entity-instance datum is linked
 to a second entity-instance datum by a specific
 relation-instance datum so that [the] meaningful
 interpretation of these two sets of tables is provided
 [u]by schema definitions stored in another finite set of
 [u]tables (RelDef and EntDef)].

11. (Amended) A machine-implemented database system comprising:

(a) entity recording means for recording data representative at least of a first entity instance belonging to a first entity class and a second entity instance which belongs either to the first entity class or a second [entity] entity class;

(b) relation recording means, logically linked to the entity recording means, for recording first data identifying the first entity instance, second data which is opposed to the first data and identifies the second entity instance and third data which is opposed to the first and second data and identifies one or more distinct relations which link the first entity instance to the second entity instance; and

(c) relation reporting means, operatively coupled to the relation recording means, for examining data recorded in the [realtion] relation recording means according to a search algorithm based on one or two of the first through third data, for identifying the opposed two or one of the first through third data and for reporting the identity of said opposed two or one.--

[Claim 12 continues to read as:]

12. The system of Claim 11 wherein the entity recording means comprises:

(a.1) a plurality of Entity-instances (Ei) tables each for recording data representative of entity instances which belong to an entity class associated with the Ei table; and

(a.2) an Entity-class defining table for recording entity-class data representative of distinct entity classes and opposed to such entity-class data, EiT data identifying one Ei table where all instances of the corresponding entity-class are recorded.

[Please amend Claims 13-14 as indicated]

AA13. (Amended) The system of Claim 12 wherein the relation recording means comprises:

(b.1) a plurality of Relation-instances (Ri) tables each of recording data representative of the distinct relations between the first and second entity instances and for associating each of the relations with a distinct relationship class; and

(b.2) a Relation-class defining table for recording relation-class data representative of each relation-class data representative of each relation class and opposed to such relation class data, RiT data identifying [identifying] one Ri table where all instances of the corresponding relation-class are recorded.

14. (Amended) A method for defining in a computerized database system, relations between [entities] entities, the method comprising:

recording first data defining a first entity class;

recording second data defining a second entity class;

recording third data defining a first group of entity instances belonging to the first entity class;

recording fourth data defining a second group of entity instances belonging to the second entity class; and

recording fifth data expressly defining a relationship instance and connecting said relationship instant to an instance of the first entity class and to an instance of the second entity class. TM

[Please add the following new Claims 15-22 as shown.]

15. With respect to a data processing system according to Claim 1, 2, 3 or 6, a database navigation method which detaches relationship navigation from function specific programming code, said method comprising the steps of:

formulating path descriptions external to program code, where the path descriptions describe how to follow pre-defined relationships from one entity type to another such that database navigation will adhere to a set of pre-defined allowed relationships;

saving the path descriptions in an inquiry-definition table (IngDef) defined in the data storing means so that the saved path descriptions may be subsequently selected for automatic execution.

16. The method of Claim 15 wherein path descriptions are interpreted as data variables rather than program source code and each path description is identified by a unique name.

17. The method of Claim 15

wherein a predefined same path description is obeyed multiple times while a plurality of paths are navigated, said plural paths defining relational connections between sets of entity-instances at the beginnings of the plural paths and sets of entity-instances at the ends of the plural paths and

wherein the navigational path restrictions of said same path description are obeyed (interpreted) during each individual path execution.

18. The method of Claim 15 wherein the InqDef table is alterable so that any original path description may be altered to overwrite the original version stored on the data storage device.

19. The method of Claim 18 wherein a pre-defined path description is altered and subsequent executions obey the new navigational description without need for re-programming of object or source code.

20. The data processing method of Claim 15, further comprising the step of allowing individual path descriptions to have their own access security independent of other path descriptions, and to be attached to separate menus.

21. A data processing system according to Claims 1, 2, 3 or 6 further comprising:

an inquiry-defining table for defining an inquiry path to be followed through said entity-instance data storing and relation-instance data storing tables.

22. A method for defining in a computerized database system, relations between entities, the method comprising:

recording first data defining a first entity class;

recording second data defining a second entity class;

recording third data defining a first group of entity instances belonging to the first entity class;

recording fourth data defining a second group of entity instances belonging to the second entity class;

recording fifth data defining an allowed relationship type; and

recording sixth data expressly defining a relationship instance and connecting said relationship instance to an instance of the first entity class and to an instance of the second entity class. *TM*